**NSA**

**Install virtualization software**

Installed VirtualBox

Created new instance of a virtual machine:

A computer screen shot of a computer screen

AI-generated content may be incorrect.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

A computer screen shot of a computer

AI-generated content may be incorrect.

A computer screen with a computer screen

AI-generated content may be incorrect.

I downloaded an Ubuntu image to use as the OS on my vm.

**Install and configure services**

For being able to use SSH and some of the other services, it is needed to go to the settings of the vm and change form NAT mode (default) to bridge node because if the vm is configured in NAT mode, it will receive its ip from the virtual box but with bridged mode it will receive directly from the main network (to which the host machine is also connected) so it will be easy to be accessed by other devices because it will act like a separate machine.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* SSH

Install ssh: use first „sudo apt update” to be sure the system got the latest packages form Ubuntu, then for the installation use „sudo apt install openssh-server”

Enable the ssh server by: „sudo systemctl enable ssh”

Start it by: „sudo systemctl start ssh” and check again if it is running with „systemctl status ssh” (it should be using port 22)

Find the ip address of the VM: „ip a” and connect from the host machine to the VM using ssh

Run in command prompt „ssh username@ip” with username being ubuntu username (check Ubuntu username with „whoami”)

* WEB

„sudo apit upgrade”

„sudo apt install apache2” to install web server (to handle http requests)

„sudo systemctl enable apache2” to enable the server

„sudo systemctl start apache2” to start it

„systemctl status apache2” to check if it runs and „ss -tlnp | grep :80” if it is running on port 80

„<http://vm_ip>” in windows browser to check if it find the web server hosted by the vm

* FTP

„sudo apt update”

„sudo apt install vsftpd” to install ftp daemon server

„sudo systemctl enable vsftpd”

„sudo systemctl start vsftpd”

„systemctl status vsftpd” to check if it is running

„sudo nano /etc/vsftpd.conf” to open the config file with editor

Inside config file:

„anonymous\_enable=NO # disable anounymous login

local\_enable=YES

write\_enable=YES”

„chroot\_local\_user=YES # optional to restrict users to their home dir

allow\_writeable\_chroot=YES”

„pasv\_enable=YES # at the end of the file to enable passive mode for bridged networks

pasv\_min\_port=40000

pasv\_max\_port=50000”

„userlist\_enable=YES # limit access to some users

userlist\_file=/etc/vsftpd.userlist

userlist\_deny=NO”

„sudo systemctl restart vsftpd” restart the ftp for applying the new configuration

„sudo adduser ftpuser” create ftp user „sergiu\_goian” and set a password: „password”

„sudo mkdir -p /home/ftpuser/ftp” create a dedicated dir

„sudo chown ftpuser:ftpuser /home/ftpuser/ftp” change ownership of file,

„sudo chmod 750 /home/ftpuser/ftp” provide owner all perm and for the user group read + execute without write

„sudo ufw allow 21/tcp”

„sudo ufw allow 40000:50000/tcp”

„sudo ufw reload”

„sudo systemctl status vsftpd” check if it is running

„ftp localhost” and then give ftp username and password to check iff active, then „ls” to see if you can see ftp dir, then „bye” to exit

Test connection from host machine: type in file explorer in address bar „<ftp://vm_ip>”, then give user and pwd

Open command prompt: ftp <vm\_ip> and login

Or can also test using file zilla

Test from windows command: „ftp vm\_ip”

* DHCP

„sudo apt install isc-dhcp-server” to install the dhcp server on vm

„sudo systemctl enable isc-dhcp-server” to enable it

„sudo systemctl start isc-dhcp-server” to start the service

„sudo nano /etc/dhcp/dhcpd.conf” to open config file

„sudo nano /etc/default/isc-dhcp-server” edit default config

* DNS
* MAIL (port 25)
* POP3/IMAP
* Pop3S/IMAPS
* proxy
* docker
* etc

**Administration tasks**

* monitor services availability (up or down)
* performance testing
* log analyzer
* backup tasks

**Security tasks**

* firewall
* brute force detection/prevention